CLAIM OR CLAIMS

WHAT IS CLAIMED IS:

1. A method of learning a finite automaton of a protocol implementation comprising the steps of:

a) grouping times within an example communication together as equivalence classes; and

b) using the equivalence classes as states of the finite automaton.

2. The method of claim 1 wherein the example communication includes PDU (Protocol Data Unit) types and the grouping step comprises the steps of:

calculating a similarity value between every two times within the example communication to form a similarity matrix, the similarity values being dependent on the length of the PDU type sequence which is coincident for and surrounds both times; and

forming the equivalence classes from the similarity matrix.

3. The method as claimed in claim 2 wherein the forming step comprises the step of transforming the similarity matrix into an equivalence matrix by means of a lower threshold for the similarity values such that the similarity values are converted into states and the times are grouped together by state to form the equivalence classes.

4. The method as recited in claim 2 wherein the forming step comprises the steps of:

forming a transitive hull for the similarity matrix between two times within the example communication to calculate an equivalence relation; and

obtaining the equivalence classes from the equivalence relation.

5. The method as recited in claim 1 wherein the using step comprises the step of entering each PDU (Protocol Data Unit) of the example communication as a state transition of the finite automaton, the state transition being a transition from the state whose equivalence class includes the time immediately prior to the PDU to the state whose equivalence class includes the time immediately after the PDU marked with the PDU type.

6. The method as recited in claim 2 further comprising the step of performing the preceding steps several times for overlapping partial sections of the example communication, with the similarity matrix of two overlapping partial sections each being united to form the similarity matrix for the example communication.

7. A method of learning a finite automaton of a protocol implementation using an example communication, the finite automaton having basic

protocol states and the state transitions of the finite automaton being marked with an appropriate Protocol Data Unit (PDU) type, comprising the steps of:

a) defining times in the example communication between every two PDUs which occur in sequence;

b) calculating a similarity value between every two times as defined in a) to form a similarity matrix, which similarity value indicates the sum of the number of PDU types coincident for and surrounding both times;

c) transforming the similarity matrix to an equivalence matrix by means of a lower threshold for the similarity values calculated according to b), such that two times fulfill an equivalence relation for an equivalence matrix if the similarity values between these two times is larger than or equal to the lower threshold;

d) forming a transitive hull for the equivalence matrix defined according to c), the transitive hull constituting equivalence classes on times according to a);

e) defining each equivalence class of the equivalence relation according to d) as a state of the finite automaton;

f) entering the PDUs of the example communication as state transitions of the finite automaton, the state transitions being a transition from the state whose equivalence class according to e) includes the time immediately prior to the PDU in question to the state whose equivalence

class according to e) includes the time immediately after the PDU in question marked with the PDU type of the PDU in question, with transitions that are identical as far as starting and sequential states and PDU type are concerned being only entered once.

8. The method as recited in claim 7 wherein steps a) to f) are performed several times for overlapping partial sections of the example communication, with the equivalence matrices according to c) of two overlapping partial sections each being united and the state-forming equivalence matrix being calculated in analogy to d) by means of the union of the equivalence matrices.

9. A method of learning arithmetic classification rules for features from a training set having positive examples comprising the steps of:

a) forming derived features on the basis of statistical measures in the form of arithmetic terms; and

b) formulating logic conditions on numerical values of the group consisting of the features from the training set and the derived features.

10. The method as recited in claim 9 wherein the forming step comprises the step of forming the derived features on the basis of correlation and regression coefficients on the training set for each possible pair of features, with the value of the derived feature being calculated from two features

from the training set or from one feature from the training set and a constant.

11. The method as recited in claim 10 wherein the formulating step comprises the step of taking the conspicuous accumulations of the values of a feature from the training set or a derived feature in a numerical value or within a numerical interval into consideration to establish the logic conditions.

12. The method as recited in claim 11 wherein the conspicuous accumulation is defined in that it maximizes the quotient of the width of the smaller one of two gaps immediately adjacent to the numerical interval in which there are no values of the feature in question, and the width of the largest gap within the numerical interval in which there are no values of the feature in question.

13. The method as recited in claim 12 further comprising the steps of:

constructing plural subclasses of the training set by organizing the logic conditions in a disjunction of clauses, with one clause constituting a conjunction of one or plural logic condition(s); and

describing a subclass each of said training set.

14. The method as recited in claim 13 further comprising the step of conducting a selection of the constructed clauses for characterizing the

entire training set such that all elements, if possible, of the training set are selected by at least one of the clauses, and as many as possible of them by exactly one clause.

15. The method as recited in claim 14 wherein the training set is an example communication composed of Protocol Data Units (PDUs) of a protocol machine and the logic conditions are the rules for the numerical PDU field contents of a sequence of PDUs.

16. A method of learning arithmetic classification rules for features from a training set having exclusively positive examples, the method being used for learning rules for numerical Protocol Data Unit (PDU) field contents of a sequence of PDUs which correspond to a specific partial path in a finite automaton of protocol basic states and PDU types, comprising the steps of:

a) interpreting each component of a feature vector as the expression of an attribute, with the number of attributes present at the beginning corresponding to the dimension of the feature vector;

b) forming new, derived attributes for each possible attribute pair on the basis of correlation and regression coefficients on the training set, with the value of a derived attribute for each feature vector being calculated from already present attribute values of the feature vector, namely as a sum, product, quotient or difference of two already present attributes, or as a product of a present attribute and a constant, in the case of a

calculation from two present attributes, the attributes considered in the calculation being selected for maximum correlation with a third attribute and the arithmetic operation for maximum correlation of the derived attribute with the same third attribute, and in the case of a multiplication with a constant, an attribute with a particularly high correlation with a second attribute being multiplied by the linear regression coefficient of the attribute pair in such a manner that the resulting derived attribute corresponds numerically to the said second attribute, if possible;

c) deriving conspicuous accumulations of the values of an original attribute or an attribute according to b) that are detected in a numerical value or within a numerical interval, a conspicuous accumulation being defined in that it maximizes the quotient of the width of the smaller one of the two gaps immediately adjacent to the numerical interval in which there are no values of the attribute in question, and the width of the largest gap within the numerical interval in which there are no values of the attribute in question;

d) forming clauses based on conspicuous accumulations as defined in c), said clauses each formulating a logic condition for selecting those examples from the training set whose attribute values of a certain attribute are within a time interval determined according to the characteristics of the associated conspicuous accumulation as defined in c), with each clause being capable of representing a conjunction of plural such selection criteria for different attributes;

e) constructing a selection of the clauses according to d) for characterizing the entire training set such that all elements, if possible, of the training set are selected by at least one of the clauses, and as many as possible of them by exactly one clause.